

# Day One Assignment

**Name :** Anushka Gupta

**SRN :** PES2UG20CS060

**Class :** 5A

**Date :** 04 July 2022

[Problem Statement 1](#)

[Problem Statement 2](#)

[Problem Statement 3](#)

[Problem Statement 4](#)

## Problem Statement 1

---

Check whether the dataset in `gen1.csv` is monotonic and find correlation using the same(spearman/Pearson)

---

## Day One Assignment

Check if the given dataset `gen1.csv` is monotonic or not

```
In [1]: # importing Python Packages
import pandas as pd

# reading the dataset
data = pd.read_csv("gen1.csv")

# displaying the dataset
data
```

```
Out[1]:
```

	temp	vp	PET	rainfall
0	22.13	23.50	5.87	0.27
1	24.16	22.39	6.52	2.69
2	26.04	24.43	7.21	30.48
3	27.03	36.90	7.26	12.83
4	26.80	45.12	6.88	116.82
...	...	...	...	...
2446	26.95	44.63	5.23	156.34
2447	26.45	37.62	5.28	0.30
2448	26.01	28.74	5.70	NaN
2449	27.12	25.59	6.27	NaN
2450	28.22	28.90	6.64	11.74

2451 rows x 4 columns

```
In [2]: data['temp'].is_monotonic
```

```
Out[2]: False
```

```
In [3]: data['vp'].is_monotonic
```

```
Out[3]: False
```

```
In [4]: data['PET'].is_monotonic
```

```
Out[4]: False
```

```
In [5]: data['rainfall'].is_monotonic
```

```
Out[5]: False
```

## Problem Statement 2

Use the WEKA Explorer and justify the values

1. MCC
2. Kappa Stats
3. ROC Curve Value

For the different pre-defined datasets present under

`C:\\Program Files\\Weka-3-8-6\\data\\diabetes.arff`

**MCC** It's a **correlation** between predicted classes and ground truth.

- +1 denotes a perfect model
- -1 denotes a poor model
- 0 denotes that the classifier is no better than a random flip of a fair coin

**Kappa Statistics** is the ratio of the proportion of times that the appraisers agree (corrected for chance agreement) to the maximum proportion of times that the appraisers could agree (corrected for chance agreement).

**ROC Curve Value** are frequently used to show in a graphical way the connection/trade-off between clinical sensitivity and specificity for every possible cut-off for a test or a combination of tests. In addition the area under the ROC curve gives an idea about the benefit of using the test(s) in question.

- 0.7 to 0.8 is considered acceptable
- 0.8 to 0.9 is considered excellent
- 0.9+ is considered outstanding

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'RandomForest' with parameters: -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1. The test options are set to 'Cross-validation' with 10 folds. The classifier output shows the following results:

```

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.32 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      582           75.7813 %
Incorrectly Classified Instances    186           24.2188 %
Kappa statistic                    0.4566
Mean absolute error                 0.3106
Root mean squared error             0.4031
Relative absolute error             68.3405 %
Root relative squared error         84.5604 %
Total Number of Instances          768

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
-----
0.836   0.388   0.801    0.836   0.818     0.458   0.820    0.886   tested_negative
0.612   0.164   0.667    0.612   0.638     0.458   0.820    0.679   tested_positive
Weighted Avg.   0.758   0.310   0.754    0.758   0.755     0.458   0.820    0.814

=== Confusion Matrix ===
  a  b  <-- classified as
418 82 | a = tested_negative
104 164 | b = tested_positive
  
```

The ROC Curve Value and PRC Area is considered excellent for this data.

MCC value being +0.450 suggests that the model is fairly good.

## Problem Statement 3

Calculate Mean, Median and mode for columns rainfall, temp, VP, PET in R

Columns	Mean	Median	Mode
Rainfall	149.5608	78.12	
Temp	25.15173	24.8	

Columns	Mean	Median	Mode
VP	48.51165	46.01	
PET	5.79288	5.46	

```
> data <- read.csv("gen1.csv")
> result.mean <- mean(data$temp)
> print(result.mean)
[1] 25.15173
> result.median <- median(data$temp)
> print(result.median)
[1] 24.8
>
```

```
> result.mean <- mean(data$vp)
> print(result.mean)
[1] 48.51165
> result.median <- median(data$vp)
> print(result.median)
[1] 46.01
>
```

```
> result.mean <- mean(data$rainfall, na.rm = TRUE)
> print(result.mean)
[1] 149.5608
> result.median <- median(data$rainfall, na.rm = TRUE)
> print(result.median)
[1] 78.12
>
```

```
Console Terminal x
G:/My Drive/Semester_5/Summer_Course/tableau/[01] dayOne/Assignment/
> data <- read.csv("gen1.csv")
> mode = function(){
+   return(sort(table(data$temp))[1])
+ }
> mode()
20.99
1
> mode = function(){
+   return(sort(table(data$vp))[1])
+ }
> mode()
10.98
1
> mode = function(){
+   return(sort(table(data$PET))[1])
+ }
> mode()
4.31
1
> mode = function(){
+   return(sort(table(data$rainfall))[1])
+ }
> mode()
0.05
1
```

## Problem Statement 4

Plot histogram for temp, vp, PET in R

```
> png(file = "temp-histogram.png")
> hist(v$temp, xlab = "Temperature", col = "green",border = "red")
> dev.off()
null device
1
> png(file = "vp-histogram.png")
> hist(v$vp, xlab = "Vapor Pressure", col = "green",border = "red")
> dev.off()
null device
1
> png(file = "pet-histogram.png")
> hist(v$PET, xlab = "Potential Evapotranspiration", col = "green",border = "red")
> dev.off()
null device
1
```





